

Appendix

This appendix is structured as follows. In **Section 1**, we provide more details on the network architecture, training setup, equipment, pose solving configuration, and data augmentation methods used in our research. In **Section 2**, we perform more ablation studies on the effects of correlation lookup in Stage 3 and the influence of the number of templates on pose precision. **Section 3** shows more qualitative results of different stages in PicoPose and more qualitative comparisons of different methods on real-world benchmarks. In **Section 4**, we conduct robotic grasping experiments in a simulation environment using the PyBullet physics engine [1] to verify the application of our method in robotic grasping. Finally, **Section 5** discusses the limitations of our approach.

We also provide **supplementary videos** demonstrating our method’s performance in two critical scenarios, including (1) *object pose estimation in open-benchmark real-world scenes* and (2) *robotic grasping under simulated environments*. For the object pose estimation video, we show our visual estimation results for each test frame in a real-world scene from YCB-Video. For the simulated grasping video, we demonstrate how to use the estimated pose for vision-guided robotic grasping. These videos show that our method is applicable to pose estimation tasks under complex visual conditions and interactive scenarios such as robotic manipulation.

1 More Experimental Details

Network architecture details. In Stage 1, we build on previous work [2] by using `dinov2_vitl14` as our backbone. The feature dimension of this backbone is 1,024 for each token. In Stage 2, we employ two conventional layers with group normalization and a ReLU activation function to reduce the spatial size of the input correspondence map \mathcal{A} to 8×8 , after which we flatten the feature map to obtain the global pose vector. In Stage 3, in order to reduce network parameters, we set the number of channels D_l of the l^{th} feature map generated from DPT [3] to 256. Additionally, we utilize the standard lookup operation in RAFT [4] for L blocks. Since the feature sizes in each block vary, the hyperparameter settings of each block need to be adjusted individually. For the l^{th} block, we establish the layer of the correlation pyramid as $l + 1$ and set the radius of the correlation lookup to 4. We also list the model sizes of different stages in Table 1.

Stage	#Param
1	304 M
2	18 M
3	58 M
Total	380 M

Table 1: The model sizes of different stages in our network.

Details of training settings. We report the detailed hyperparameter settings to train our network in Table 2.

Hyperparameters	Settings
Optimizer	AdamW
AdamW β	(0.5, 0.99)
AdamW ϵ	1e-6
Learning rate scheduler	Cosine decay
Training iterations	400,000
Warmup iterations	1,000
Learning rate	1e-5
Weight decay	5e-4
Batch size	32

Table 2: Detailed hyperparameters in training our network.

More details of devices. We conducted all experiments with GPU in GeForce RTX 3090 24G, and CPU in Intel (R) Xeon (R) CPU E5-2678 v3 @ 2.50 GHz under the Linux operating system.

Details of PnP/RANSAC. We utilize the EPnP algorithm [5] along with the RANSAC scheme in the fine correspondence to solve the object pose in Stage 3. The RANSAC iterations are configured to 150, and the reprojection error threshold is set to 2.

Details of augmentations. In Stage 1, to better adapt to the input images in real-world scenarios, we conduct data augmentation on the query image of the training data in a manner similar to GDRNPP [6]. In Stage 3, we apply random 2D translation, in-plane rotation, and scale noise to the ground truth affine transformation \mathcal{M} to generate the initial coordinate map \mathcal{P} .

2 More Ablation Studies

Effects of correlation lookup in Stage 3. In Stage 3, we use the correlation lookup operation in RAFT [4] to obtain flow features, and combine them with the features of the input image and the best-matched template to predict coordinate offsets and the certainty map. To verify the effectiveness, we conducted experiments without using the correlation lookup operation. As shown in Table 3, the features obtained by the correlation lookup operation can improve the results significantly.

Correlation Lookup	LM-O	T-LESS	YCB-V	MEAN
×	35.0	25.9	46.8	35.9
✓	46.3	39.7	58.7	48.2

Table 3: Quantitative results of correlation lookup operation in Stage 3. We report the mean Average Recall (AR) among VSD, MSSD and MSPD.

Influence of the number of templates. We follow the setup of GigaPose [2] by using $N = 162$ templates per object in our evaluation experiments. In Table 4, we present additional quantitative results with different numbers of templates for both GigaPose and our proposed PicoPose. The results show improvement as more templates are used, since both methods rely on template matching to select the best-matched template for the target object. However, the rate of improvement slows as the number of templates increases. Notably, PicoPose is more effective than GigaPose when using fewer templates, further highlighting the advantages of PicoPose.

Method	#Temp	LM-O	T-LESS	YCB-V	MEAN
GigaPose [2]	2	4.8	4.4	2.0	3.7
PicoPose		10.5	12.8	14.1	12.5
GigaPose [2]	6	11.5	9.2	5.9	8.9
PicoPose		27.5	25.0	39.5	30.7
GigaPose [2]	42	25.0	23.3	23.4	23.9
PicoPose		43.9	37.9	57.4	46.4
GigaPose [2]	162	29.6	26.4	27.8	27.9
PicoPose		46.3	39.7	58.7	48.2

Table 4: Quantitative comparison with GigaPose [2] on the number of templates. We report the mean Average Recall (AR) among VSD, MSSD and MSPD.

3 Additional Qualitative Results

More qualitative results of different methods. We present more qualitative results of different methods on the seven core datasets (LM-O[7], T-LESS[8], TUD-L[9], IC-BIN[10], ITODD[11], HB[12], and YCB-V[13]) in the BOP benchmark [14], shown in Fig. 1. We illustrate the estimated 6D pose by rendering the 3D model on the input image and using the overlap rate as a basis, where a higher overlap rate indicates a more accurate estimated 6D pose. Specifically, all methods use the same zero-shot segmentation method, i.e., CNOS [15].

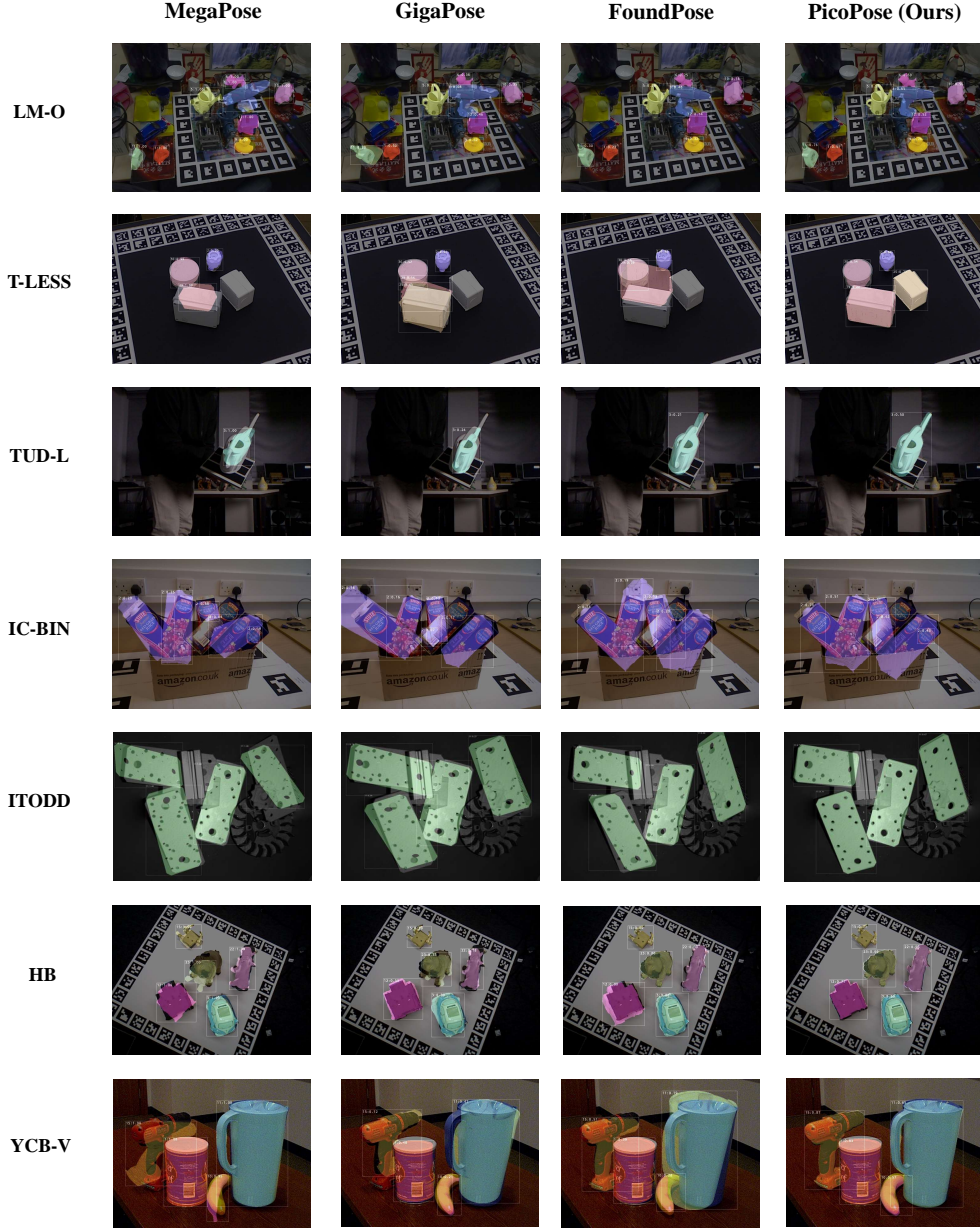


Figure 1: More qualitative results of different methods on the seven core datasets of BOP benchmark [14], including LM-O, T-LESS, TUD-L, IC-BIN, ITODD, HB, and YCB-V, arranged from top to bottom.

59 **More qualitative results of different stages.** We visualize the correspondences between the query
60 image \mathcal{I} and the best-matched template \mathcal{T} in Stage 1 and Stage 2. As shown in Fig. 2, the coarse
61 correspondences generated in Stage 1 contain numerous outliers and inconsistencies, many of which
62 are effectively resolved by Stage 2 to produce smooth correspondences. In Stage 3, we enhance
63 the display of fine correspondences by visualizing the coordinate map \mathcal{P} as optical flow with the
64 certainty map on YCB-V dataset [13], shown in Fig. 3.

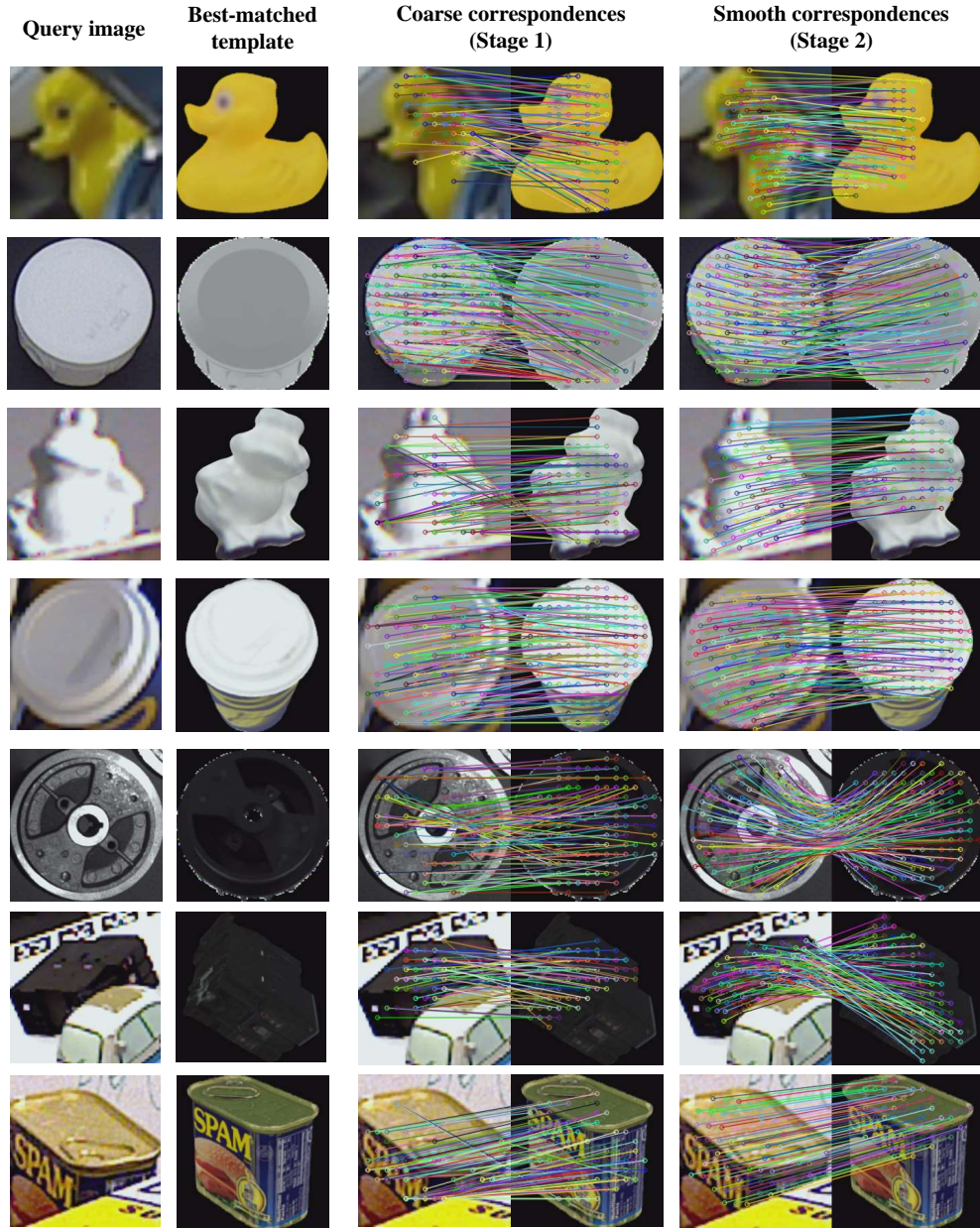


Figure 2: Qualitative results of coarse correspondences in Stage 1 and smooth correspondences in Stage 2 on the seven core datasets of BOP benchmark [14], including LM-O, T-LESS, TUD-L, IC-BIN, ITODD, HB, and YCB-V, arranged from top to bottom.



Figure 3: Qualitative results of fine correspondences in Stage 3 on YCB-V dataset [13].

4 Application: Robotic Grasping in Simulated Environments

In this section, our proposed PicoPose demonstrates seamless integration for robotic grasping applications using PyBullet [1] with the setup shown in Fig. 4. Our experimental scene comprises (1) a Franka Emika Panda robotic arm, (2) distractor objects, including the target, randomly arranged on the workspace, and (3) a placement tray. A fixed virtual camera captures single RGB images of the cluttered scene as input to our system.

The processing pipeline consists of three key stages. First, CNOS [15] segments the target object in the RGB scene. Second, our proposed PicoPose estimates the 6D pose of the target object in camera coordinates. Third, we use the known camera-to-robot coordinate transformation to convert this pose into a 6D grasping pose and use inverse kinematics to generate robot motions to successfully grasp the target object. Complementary to Fig. 4, supplementary videos demonstrate the complete grasping process in action.

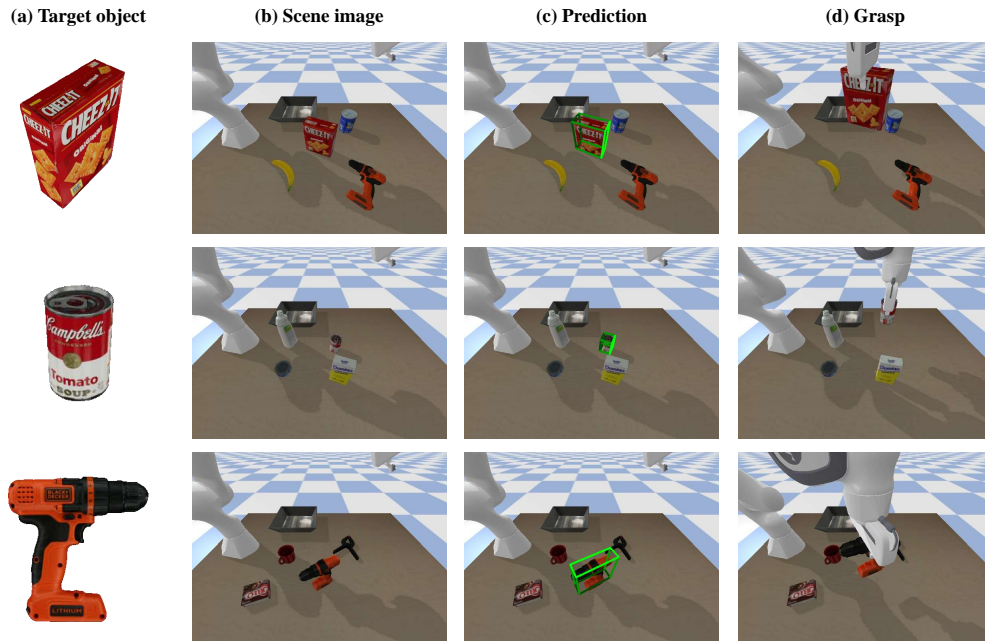


Figure 4: Robotic grasping application of PicoPose in simulated environment. The complete grasping process is demonstrated in the supplementary videos.

5 Limitations

While PicoPose demonstrates strong performance on 6D object pose estimation benchmarks with real-world cluttered scenes and shows promising potential for rapid deployment in robotic applications through simulated grasping experiments, several areas remain for future improvement. First, the reliance on multiple templates for 3D object representation means that its performance is inherently tied to the number of templates used. Although our approach achieves comparable results with fewer templates than existing methods (as detailed in Supplementary Section 2), future work could explore more efficient template utilization strategies or alternative 3D representations. Second, while PicoPose benefits from the iterative refinement of MegaPose [16], the performance gains are relatively modest compared to other methods, probably because PicoPose’s initial estimates already approach the refiner’s performance upper bound. This motivates the development of specialized refinement approaches for high-quality initial predictions. Third, in case of single image reference of the target object, where we follow GigaPose [2] to use Wonder3D for reconstructing the object

in 3D space, the quality of reconstruction remains a limiting factor that affects final pose estimation accuracy. We believe integrating emerging neural reconstruction techniques could significantly improve performance in this challenging scenario.

References

- [1] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. <http://pybullet.org>, 2016–2019.
- [2] V. N. Nguyen, T. Groueix, M. Salzmann, and V. Lepetit. Gigapose: Fast and robust novel object pose estimation via one correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9903–9913, 2024.
- [3] B. A. K. V. Ranftl, René. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.
- [4] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.
- [5] V. Lepetit, F. Moreno-Noguer, and P. Fua. Ep n p: An accurate o (n) solution to the p n p problem. *International journal of computer vision*, 81:155–166, 2009.
- [6] X. Liu, R. Zhang, C. Zhang, B. Fu, J. Tang, X. Liang, J. Tang, X. Cheng, Y. Zhang, G. Wang, and X. Ji. Gdrnpp. https://github.com/shanice-l/gdrnpp_bop2022, 2022.
- [7] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part II 13*, pages 536–551. Springer, 2014.
- [8] T. Hodaň, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.
- [9] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrike, X. Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.
- [10] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3583–3592, 2016.
- [11] B. Drost, M. Ulrich, P. Bergmann, P. Hartinger, and C. Steger. Introducing mvtec itodd-a dataset for 3d object recognition in industry. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 2200–2208, 2017.
- [12] R. Kaskman, S. Zakharov, I. Shugurov, and S. Ilic. Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.
- [13] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.
- [14] T. Hodan, M. Sundermeyer, Y. Labbe, V. N. Nguyen, G. Wang, E. Brachmann, B. Drost, V. Lepetit, C. Rother, and J. Matas. Bop challenge 2023 on detection segmentation and pose estimation of seen and unseen rigid objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5610–5619, 2024.

- 133 [15] V. N. Nguyen, T. Groueix, G. Ponimatkin, V. Lepetit, and T. Hodan. Cnos: A strong base-
134 line for cad-based novel object segmentation. In *Proceedings of the IEEE/CVF International*
135 *Conference on Computer Vision*, pages 2134–2140, 2023.
- 136 [16] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier,
137 M. Aubry, D. Fox, and J. Sivic. Megapose: 6d pose estimation of novel objects via render
138 & compare. *arXiv preprint arXiv:2212.06870*, 2022.